

Assignment #3
CSCI 201 Spring 2017
4.5% of course grade

For this assignment, you will continue practicing your knowledge of html, css, jsp, java and xml. In addition, you will now be incorporating javascript into your program.

This assignment will focus on expanding the UI, and adding functionality to

- create new users
- allow users to follow/unfollow other users
- allow users to perform actions on movies

XML Changes

First of all, there will be another alteration/addition to the xml file. Now, a movie object will also contain an image tag that links to the movie poster image. In addition, the actors tag will no longer be as simple as a list of names. Each actor will now be it's own object, with the first name, last name, and a URL to an image of them.

In addition, instead of simply storing the average rating of a movie, we will store both the total rating (<rating-total>), and the number of people that have rated it (<rating-count>). In order to acquire the average rating, you would divide the total by the count.

```
<movie>
    <title>How to Train Your Dragon</title>
    <director>Dean DeBlois</director>
    <image> http://www.imdb.com/title/tt0892769/
mediaviewer/rm2210040832</image>
    <writers>
        <writer>Dean DeBlois</writer>
        <writer>William Davies</writer>
        <writer>Chris Sanders</writer>
    </writers>
    <year>2010</year>
    <genre>Animated</genre>
    <description>A hapless young Viking who aspires
to hunt dragons becomes the unlikely friend of a young dragon
himself, and learns there may be more to the creatures than he
assumed.</description>
    <rating-total></rating-total>
    <rating-count></rating-count>
    <actors>
        <actor>
```

```

        <fname>Jay</fname>
        <lname>Baruchel</lname>
        <image> http://www.imdb.com/name/
nm0059431/mediaviewer/rm2699410176</image>
        </actor>
        <actor>
            <fname>Gerard</fname>
            <lname>Butler</lname>
            <image> http://www.imdb.com/name/
nm0124930/mediaviewer/rm1194363136</image>
            </actor>
        </actors>
    </movie>

```

You can assume that every movie and actor will have a valid URL in their image tag, and if a movie does have an average rating, both the total and count fields will be populated (with integers).

Data Persistence

Most of the functionality that you will be adding to the application will require you to save information back to the xml file. Whether you chose to use DOM or SAX in your previous assignments, you will need to add code to write back to the file. This will be required of you when:

- A user rates a movie, this should be added to the <rating-total> value, and the <rating-count> value should be incremented.
- A user performs any action on a movie, this should be added as an event in their feed.
- The logged in user follows another user, that user should be added to their following list.
- The logged in user unfollows another user, that user should be removed from their following list.
- A new user is created, this user should be added to the list of users.

If any of these actions are performed, the xml file should be modified such that if the program is terminated and ran again (with the same file chosen), any changes that were made previously are reflected on the website.

Front End Additions

In order to support these new functionalities, as well as improve the overall aesthetics of the website, we will be modifying existing pages and including more pages as well.

There are a lot of subtleties involved with each page, so please read the instructions carefully, and reread them before posting a question on Piazza!

In addition, similar to last assignment, you will be graded on the aesthetics of your front end. You should aim for the underlying architecture to match the provided screenshots (i.e data is organized in the same way, elements are in the same relative places on the screen), but you can and should develop your own design using colors, fonts, borders, etc.

Menu Bar Header

The normal menu (with the six options) will now be displayed on the top of every page in the form of a menu bar. For this reason, when a user logs in, they should be navigated to their feed (similar to when you log into Facebook).

The menu bar should have an exit button, logout button, profile button, and feed button. In addition, it will have a search bar. In the example below, the search bar has the default text 'Search movies' with a clapperboard icon next to it. (A clapperboard is the object used to indicate what take the director is on when filming a movie or show). This means that the search bar is currently set to search by movies. However, if the user were to click the clapperboard icon, this would toggle the search bar to now search by users instead (shown in Figure 2).

In order to execute a search, the user must click the magnifying glass icon after entering their search parameter.



Figure 1: Menu Header
(from left to right) Feed button, profile button, search toggle button, search button, logout button, exit button



Figure 2: Menu Header, search toggled to user search

Feed

For a user with a non-empty following list, their feed should be populated with the events from those they are following (but **not** display their own events). Each event should follow the same pattern: the profile picture of the involved user, then an icon displaying what action they took, then the movie poster image.

For the 'liked' action, a thumbs up should be displayed. For the 'disliked' action, a thumbs down should be displayed. For the 'watched' action, an old school tv should be displayed. Finally, for the 'rated' action, stars should be displayed, with the number of stars corresponding to the rating the user gave (In the example below, Emma gave a 5 star rating to 'Catch Me If You Can').

Note: the rating value in an event is out of ten, but to give a cleaner look we will use 5 stars (divide the user's rating by 2, and round up to get this value).

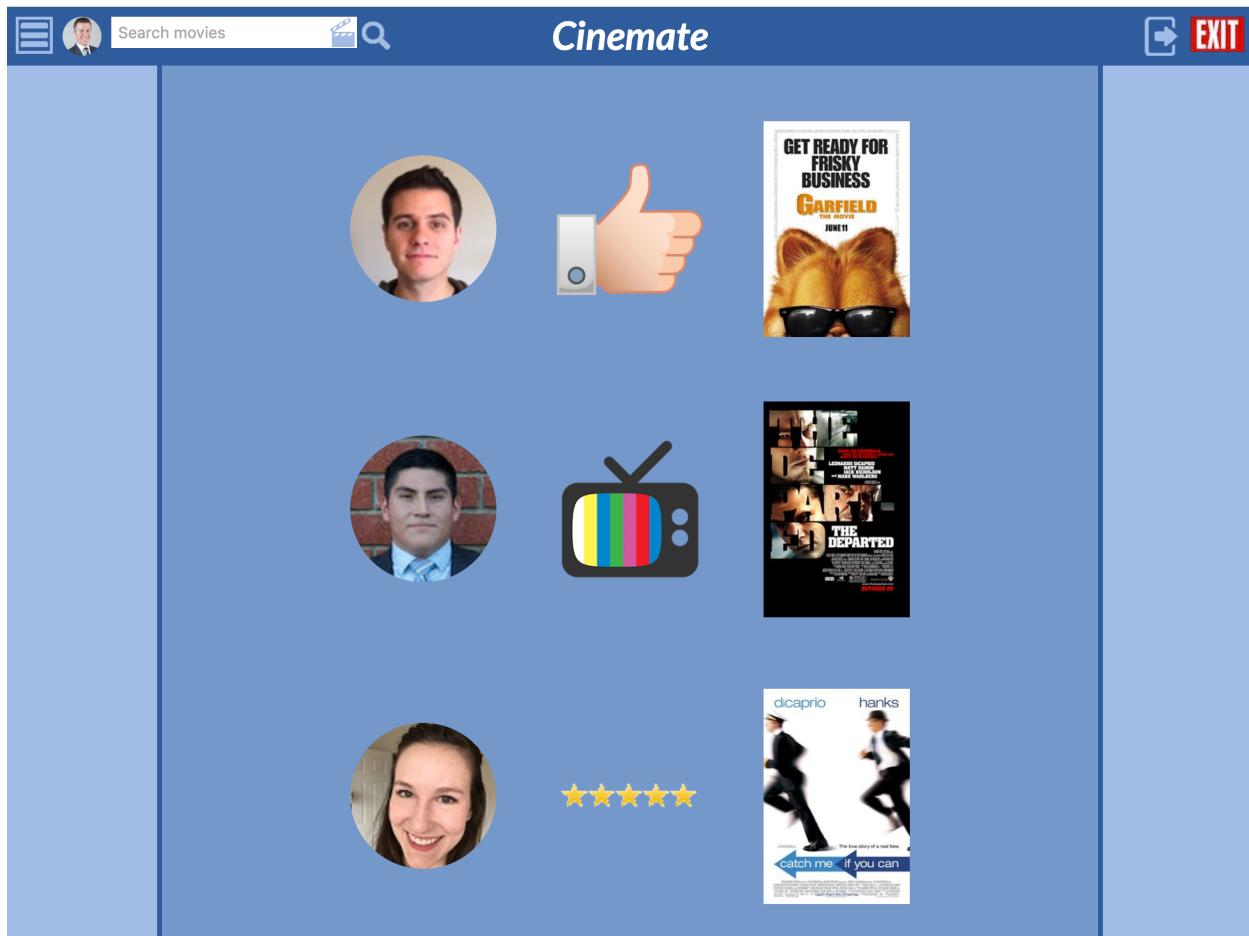


Figure 3: User's feed

The profile picture of the involved user should be clickable, and should navigate to their profile page (where the logged in user will have the option to follow/unfollow them) (Figure 5).

Similarly, the movie poster should be a clickable link that navigates the logged in user to a page with details about the movie (Figure 7).

Both the profile page and movie info page will be discussed in more detail further on in the instructions.

Searching

Rather than using the search page, and the movie search menu page, we will be utilizing the search bar. When the logged in user clicks the search icon, they should be navigated to the search page from Assignment 2. However, you will need to eliminate the search bar that was originally on the page since it is already included in the menu header.

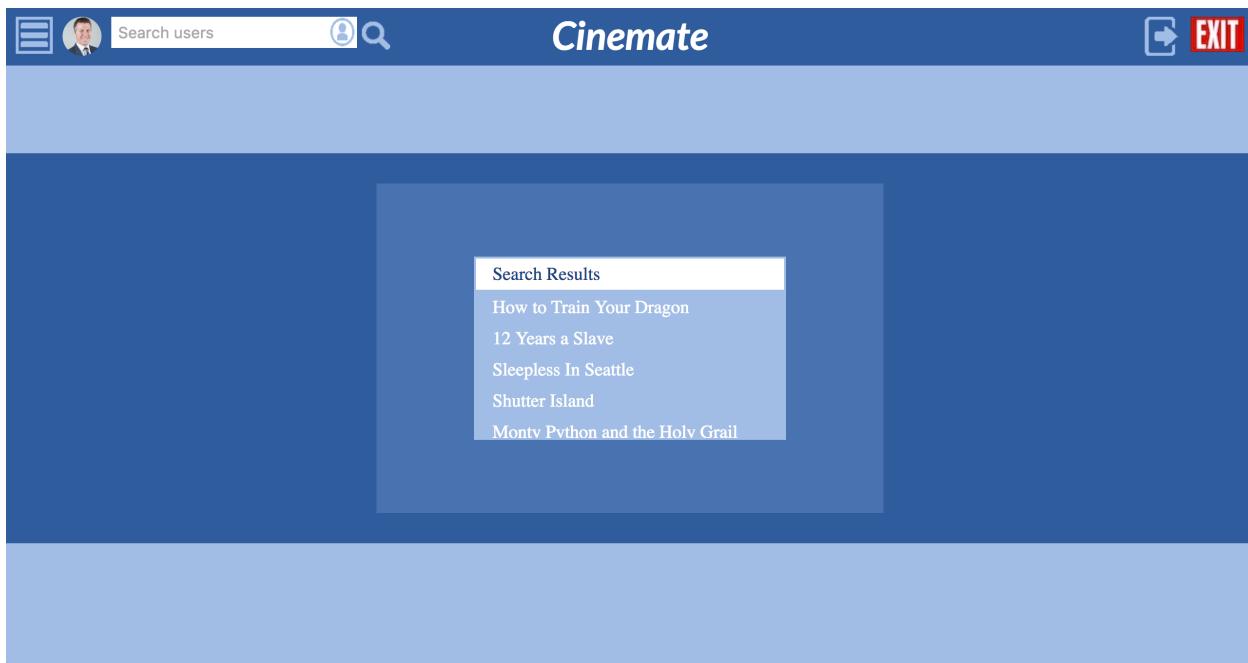


Figure 4: Search results page

Search bar above results has been removed, menu bar header has been added

Now that we don't have a submenu for searching for movies, we need some way to determine which search we are doing. So, when the search bar is toggled to movie search, the user must enter a search key that precedes their search parameter. These keys include 'actor:', 'genre:' or 'title:', which will search by actor, genre or title,

respectively. You can assume a user will always input a valid, non-empty, search key with their search parameter.

For example, in order to search for all comedy movies, the user should toggle to the movie search and enter 'genre:comedy' (or 'genre: comedy' should also work). In order to search for all movies with Will Smith, the user would enter 'actor:Will Smith' (or 'actor: Will Smith' should also work).

The results should now should be clickable links. If the results are usernames, each of them when clicked should navigate to that user's profile page. If the results are movie titles, each of them when clicked should navigate to that movie's info page. Both of these pages will be described further on.

Profile page

The profile page is essentially the same as from last assignment but with a couple modifications. It will of course have the menu header now, but instead of two columns (one for following and one for followers), it will now have three. The middle column will now display the events from only their feed. For example, if we were looking at Jeff's profile page, we would only see events with actions that he performed (ex: jvmclean watched 'Garfield: The Movie').

Each event should follow the same pattern: An icon that corresponds to the action that was taken, the action that was taken, and then the movie title. If the action was 'Rated', the icon should display the number of stars corresponding to the user's rating of the movie.

The profile page will be the same whether or not the logged in user is viewing their own profile or someone else's. The only difference will be that if they are viewing someone else's profile, there should either be a 'Follow' or 'Unfollow' button. If the user is in their following list, an 'Unfollow' button should be shown; otherwise a 'Follow' button should be shown.

If the logged in user chooses to follow the user, this update should be visible on both their profile page and their feed. In other words, if they navigate back to their profile page, they should see this new username in their following list, and if they navigate to their feed, they should now see any events from the recently added user's feed. In addition, the profile page being viewed should update to show the logged in user in the followers list. **You may not reload the entire page in order to display a change in the followers list, only the followers column should update (you can utilize AJAX to achieve this).**

Followers

- jmiller
- jvmclean

Following

- hgil
- elautz
- mtanaka
- smurmu
- ajiang
- knguyen

Unfollow

Figure 5: Professor Miller is currently following Jeff
Professor Miller is listed in Jeff's followers list and an 'Unfollow' button is displayed

Similarly, if the logged in user chooses to unfollow a user, that user should be removed from their following list and feed, and the logged in user should be removed from the followers list of the user being viewed.

Followers

- jvmclean

Following

- hgil
- elautz
- mtanaka
- smurmu
- ajiang

Follow

Figure 6: After Professor Miller chooses to unfollow Jeff
Professor Miller has been removed from Jeff's followers list and a 'Follow' button is displayed

Movie Info Page

When a user clicks a link to a movie, they should be directed to a page that displays all the information about the movie. The movie poster image should be displayed on the left, and details about the movie (genre, year, director, writers, and description) should be displayed next to it on the right. Its average rating should be displayed by filling in the appropriate number of stars out of the ten empty stars displayed (see Figure 7).

The actors should be listed at the bottom of the page in a list with their headshot then full name (this is similar to a movie page on IMDB, so it may be helpful to check that out).

The image shows a movie info page for the film "Catch Me If You Can" (2002) on a website called "Cinemate". The page has a blue header with the site's logo and navigation links. The main content area features a movie poster for "Catch Me If You Can" on the left, showing Leonardo DiCaprio and Tom Hanks in suits. To the right of the poster, the movie's title "Catch Me If You Can (2002)" is displayed in large letters, followed by its genre ("Biography"), director ("Steven Spielberg"), actors ("Leonardo DiCaprio, Tom Hanks, Christopher Walken"), and writers ("Jeff Nathanson, Frank Abagnale Jr."). Below this, a star rating of four and a half stars is shown, with a brief plot summary: "The true story of Frank Abagnale Jr. who, before his 19th birthday, successfully conned millions of dollars' worth of checks as a Pan Am pilot, doctor, and legal prosecutor." At the bottom of the page, there is a "Cast" section listing the three lead actors with their names and small profile pictures: Leonardo DiCaprio, Tom Hanks, and Christopher Walken.

Figure 7: Movie info page

You will notice there are 3 icons underneath the movie poster: the 'watched', 'like' , and 'dislike' icons. If a user clicks on any of those buttons, an event should be added to their feed (which remember is displayed on their profile page).

In order for a user to rate a movie, they should be able to hover over the average rating stars. You should indicate somehow that the user is about to submit a rating. For instance, in the example below, the user hovered over the 7th star, so we removed the black outline of the first 7 stars to indicate they are about to perform a rating action.

If a user clicks, an event should be added to their feed, and the average rating of the movie should update. If their rating was significant enough to change the average rating of the movie, the stars should update accordingly.

Once again, you may not reload the entire page in order to display a change in the average rating, only the rating stars should update.



Figure 7: Mouse hovering over rating stars to perform rating action
The movie has an average rating of 8/10, and the user is about to give it a rating of 6/10

New User

The start page should now display an option to sign up a new user.

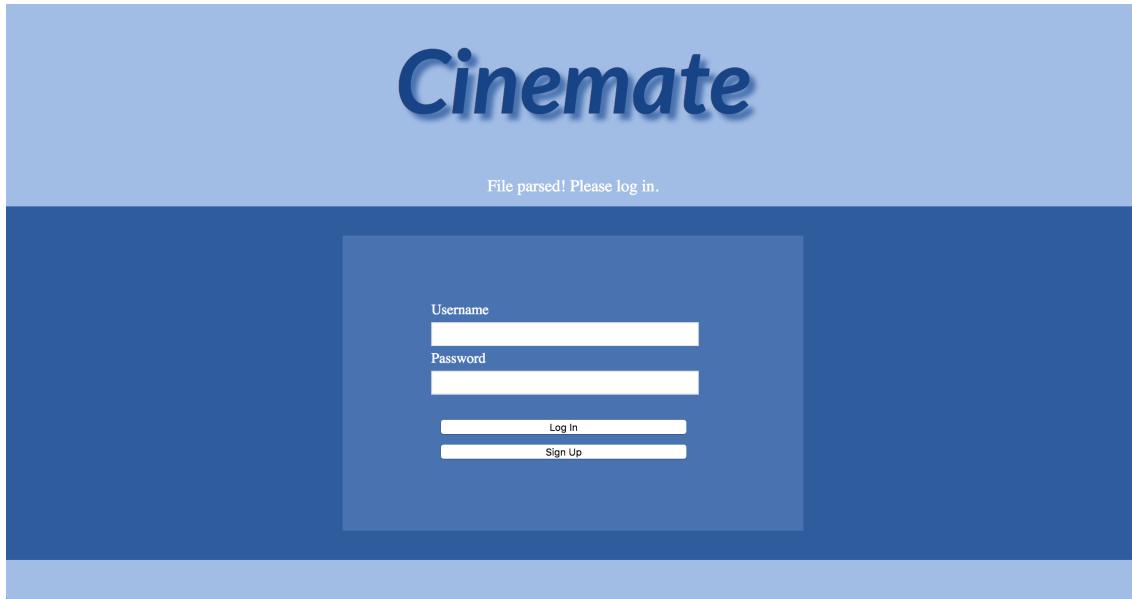


Figure 8: New user button on start page

If this option is clicked, the user should be redirected to a form to input all the necessary information. The need to provide their name, username, profile picture URL, and password.

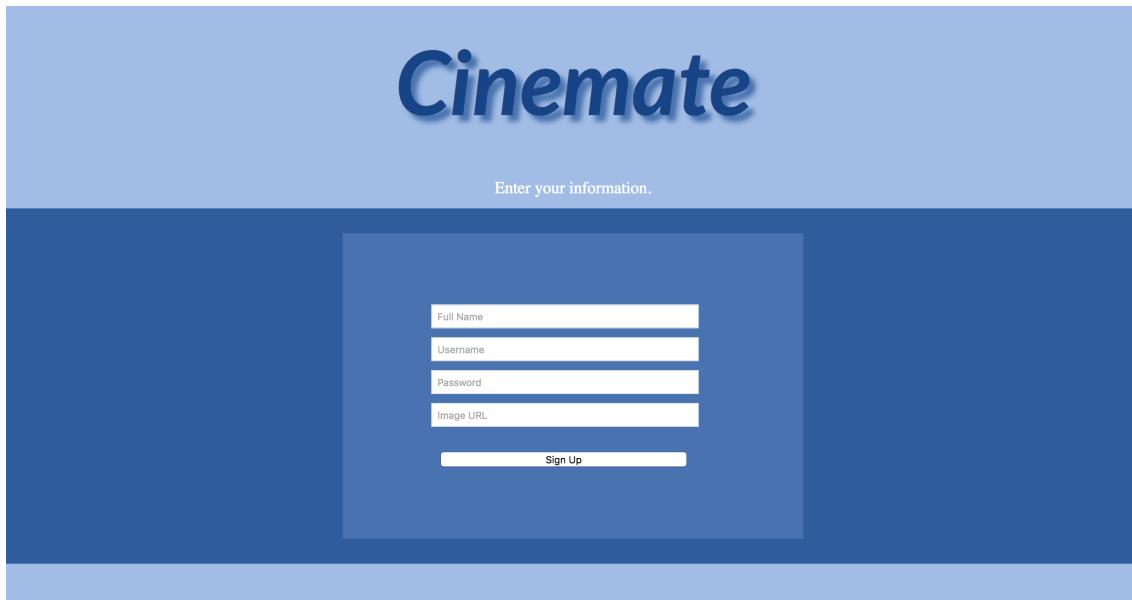


Figure 9: New user page

If they try to submit the form, but fail to provide one or more of the requested fields, an error message should appear beneath the 'Sign Up' button, stating that not all required fields were provided. If they try to submit with a username that has already been taken, an error message should appear under the 'Sign Up' button.

It will be your responsibility to parse out the first and last name of the user from the name input field. However, you can assume the user will always input a first name, then a whitespace, followed by their last name.

If they press the 'Sign Up' button with all the requested fields filled out and a unique username chosen, they should be logged in and taken to their feed (which will be empty at this point).

Error Messages

In order for you to practice AJAX calls, anytime you need to display an error message, you may not reload the entire page. Instead, you must remain on the page and only update the error message field.

This applies for the entry page (when you are trying to parse a file), the start page (when you are trying to login) and the create new user page (when they fail to provide all fields or choose an invalid username).

Tooltips

For most of these pages, we will have tooltips to make it easier to navigate the buttons. A tooltip is a label that displays when your mouse hovers over an icon/button to give more detail about the element.

For the menu bar, all of the buttons should have a tooltip.



Figure 10: Menu bar header

The movie info page should show a tooltip over each action and over the movie poster.

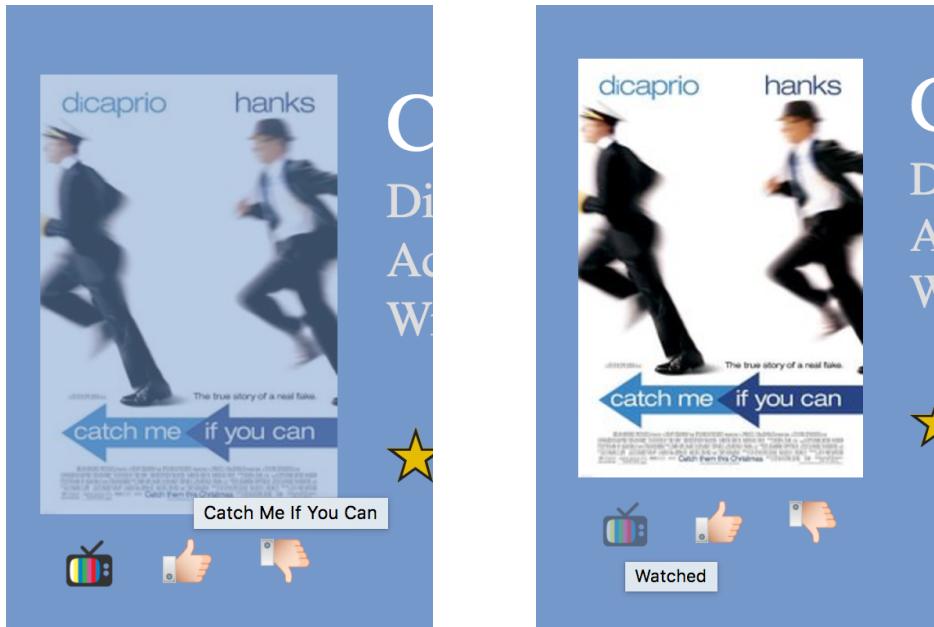


Figure 11: Movie info page tooltips

For each event in the feed, there should be a tooltip for the user, a tooltip for the action, and a tooltip for the movie.

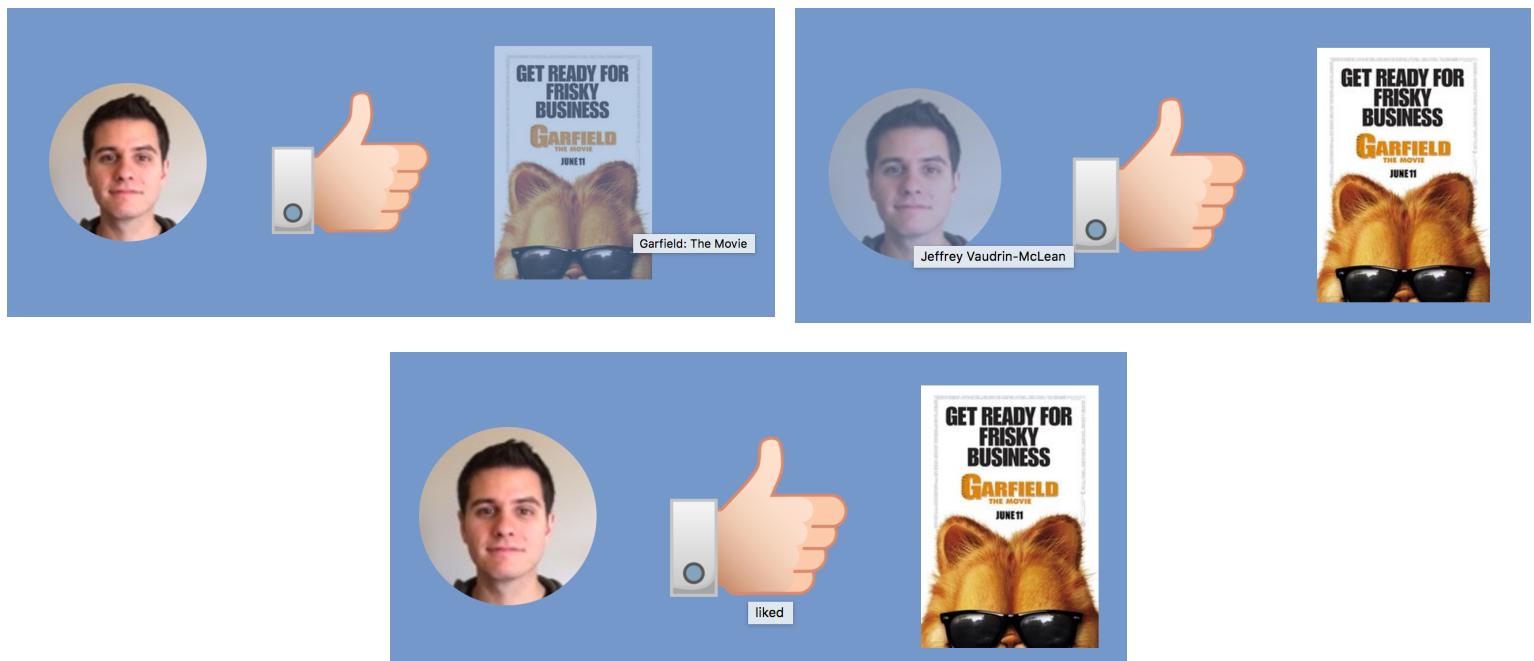


Figure 12: Event tooltips

Grading Criteria (4.5%)

Tooltips (0.1%)

- 0.05% All icons on the menu header have tooltips
- 0.025% The action, movie and user in each feed event have tooltips
- 0.025% The actions and movie poster on the movie info page have tooltips

Error Messages (0.15%)

- 0.05% The entry page does not reload in order to display an error message
- 0.05% The start page does not reload in order to display an error message
- 0.05% The new user page does not reload in order to display an error message

New User (0.1%)

- 0.025% An error is shown if the user tries to submit a form that is missing one of the fields
- 0.025% An error is shown if the user tries to submit a form with an already taken username
- 0.05% The front end for the new user looks similar to the provided screenshot (Figure 9)

Writing to XML (0.7%)

- 0.1% If a new user is created, they are added to the list of users in the xml file
- 0.2% If a user performs an action on a movie object, that is added as an event in their feed in the xml file
- 0.2% If the user rates a movie, this accurately modifies the <rating-total> and <rating-count> fields of that movie in the xml file
- 0.1% If the logged in user follows a new user, that user is added to their following list in the xml file
- 0.1% If the logged in user unfollows a user, that user is removed from their following list in the xml file

Movie Info Page (0.9%)

- 0.4% The movie info page looks similar to the provided screenshots (Figure 7)
- 0.2% The movie info page accurately displays all of the movie information
- 0.1% If the user rates the movie, the average rating stars are updated accordingly on the front end without reloading the whole page
- 0.2% If the user performs any action on the movie object, that user's feed (displayed on their profile page) is updated accordingly

Profile Page (1.2%)

- 0.35% The profile page looks similar to the provided screenshots (Figure 5)
- 0.25% The profile page now displays another column with the events from that user's feed (and only their feed)
- 0.1% If the profile page does not correlate to the logged in user, there is a follow or unfollow button displayed, depending on whether they are already following them
- 0.15% If the logged in user chooses to follow or unfollow the user, the followers list on that user's profile page should update accordingly without reloading the whole page.
- 0.05% If the logged in user chooses to follow or unfollow the user, their following list on their own profile page should update accordingly
- 0.1% If the logged in chooses to follow or unfollow another user, their feed page should update accordingly.
- 0.1% The usernames in the following and followers list are clickable, and navigate to the correct user's profile page.
- 0.1% The movie titles displayed in event event in the user's feed are clickable, and navigate to the correct movie info page.

Feed Page (0.65%)

- 0.2% The feed page looks similar to the provided screenshots (Figure 3)
- 0.1% The feed page shows the events from all the users the logged in user is following
- 0.025% The feed page does not show the events from the logged in user
- 0.125% Each event shows the user's profile picture, the correct action icon, and the movie poster
- 0.1% The profile picture of each event is clickable, and navigates to that user's profile page
- 0.1% The movie poster image of each event is clickable, and navigates to that movie's info page

Menu Header (0.4%)

- 0.15% The menu header looks similar to the provided screenshots (Figure 1)
- 0.03% The exit icon navigates to the entry page when clicked
- 0.03% The logout button navigates to the start page
- 0.03% The profile icon navigates to the logged in user's profile page
- 0.03% The feed icon navigates to the feed page
- 0.03% The search icon navigates to the search results page
- 0.1% The search toggle button correctly toggles between searching for users and searching for movies

Searching (0.3%)

- 0.15% When the search bar is toggled to movies, the search key is correctly handled to display either genre, title or actor search results
- 0.075% If the search is by users, all usernames in the search results should be clickable and navigate to that user's profile page.
- 0.075% If the search is by movies, all movie titles in the search results should be clickable and navigate to that movie's info page.